Caching at the Mobile Edge: a Practical Implementation

Poderys, Justas; Artuso, Matteo; Lensbøl, Claus Michael Oest ; Christiansen, Henrik Lehrmann; Soler, José

Published in:
IEEE Access

Link to article, DOI:
10.1109/ACCESS.2018.2809490

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 11, 2018, accepted February 19, 2018, date of publication February 27, 2018, date of current version March 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2809490

Caching at the Mobile Edge: A Practical Implementation

JUSTAS PODERYS, (Student Member, IEEE), MATTEO ARTUSO, (Student Member, IEEE), CLAUS MICHAEL OEST LENSBØL, (Student Member, IEEE), HENRIK LEHRMANN CHRISTIANSEN, (Member, IEEE), AND JOSÉ SOLER, (Senior Member, IEEE)

Department of Photonics Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

Corresponding author: Justas Poderys (juspo@fotonik.dtu.dk)

ABSTRACT Thanks to recent advances in mobile networks, it is becoming increasingly popular to access heterogeneous content from mobile terminals. There are, however, unique challenges in mobile networks that affect the perceived quality of experience (QoE) at the user end. One such challenge is the higher latency that users typically experience in mobile networks compared with wired ones. Cloud-based radio access networks with content caches at the base stations (BSs) are seen as a key contributor in reducing the latency required to access content and thus improve the QoE at the mobile user terminal. In this paper, a prototype implementation of a mobile edge cache system is presented. The proposal focuses on compliance with the existing long-term evolution deployment and content-location solutions. The prototype is designed to perform assessment tests and evaluation of caching solutions. Results are then shown for the QoE improvements for the mobile user obtained by caching content at the BSs. This is quantified with a comparison to noncached content by means of ping tests (10%–11% shorter times), a higher response rate for Web traffic (1.73–3.6 times higher), and an improvement in the jitter (6% reduction).

INDEX TERMS Mobile edge caching, 5G, practical implementation.

I. INTRODUCTION

The architecture and characteristics of mobile networks are going through comprehensive investigations to meet the challenging objectives depicted in 5G scenarios. The overall trend focuses on a Heterogeneous Network (HetNet) with diverse types of Base Stations (BSs), the distribution of the content at the edges and the centralization of functionalities [1].

Mobile Edge Caching (MEC) has several potential benefits, especially in terms of perceived Quality of Experience (QoE) for the users consuming the content. In fact, it is possible to reduce the latency for retrieving the content, reduce the number of network hops between the locations of the content and the one of its consumer and thus ultimately benefit from a better link between the two that reduces loading or buffering time [2]. Depending on the type of content the benefits can vary. Latency-sensitive and high-demand/high-volume contents are among those that can benefit the most from MEC. In this category we have—among others—video content that represents an ever increasing fraction of the data transferred over mobile networks [3]. Content could thus be cached all the way from its original storage at the provider's premises to the edge of the mobile network. Such edge storage location is at the BSs themselves, as can be seen in the architecture diagram in Fig. 1.

The work described here aims to assess and quantify some of these benefits, especially user-facing ones, such as web-browsing QoE. This is achieved by focusing on common performance metrics used to assess traffic carried over the Hypertext Transfer Protocol (HTTP), for its prominance in mobile networks traffic [3]. Such metrics are HTTP response rate, network delays and packet delay jitter. Improvements in these fundamental metrics eventually lead to shorter application response times, reduced interruptions in streaming multimedia, reduced buffering time and ultimately improved QoE. This is achieved by presenting a prototype implementation of a practical MEC system for Long Term Evolution (LTE)-based 5G networks.

Additionally, there has been a considerable amount of research into bringing some useful concepts from the cloud computing world into the Radio Access Network (RAN) design. These are referred to as Cloud-Radio Access Network (C-RAN) and the related (albeit different) concepts

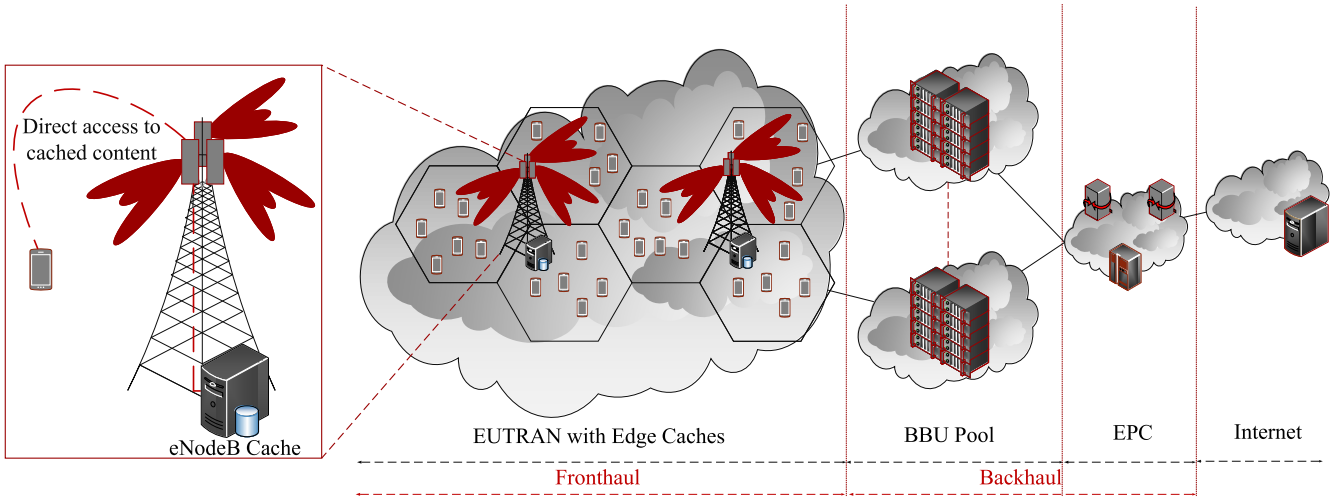

FIGURE 1. F-RAN with edge caching. Content caches are co-located with base-stations.

of Heterogeneous Cloud Radio Access Network (H-RAN) and F-RAN [4]. These architectures are proposed to enable centralization of selected network functions and signal processing. Depending on the extent of the centralization—i.e. how low the functional split in the protocol stack between central and remote units is—different characteristics in terms of capacity and latency are required for the Fronthaul (FHL) network. This is the network segment interconnecting the centralized processing pool—Baseband Unit (BBU)—and the sites where the "simplified" BSs—Remote Radio Heads (RRHs)—are located. In certain cases, even Gbps of capacity and sub-millisecond latency could be required on the FHL, thus making it a costly and demanding portion of the network.

MEC is deemed beneficial also in such architectures as pointed out in [2]. In fact, as content is moved to the BS, the hosting servers as well as the Content Delivery Network (CDN) are relieved from serving a portion of the requests. More specifically, this also implies a reduced load on the Backhaul (BHL) and the FHL, which potentially alleviates the requirements on these network portions and is thus beneficial from the perspective of the operator and infrastructure provider [2].

In the remainder of the paper, such practical implementation is contextualised in the bigger picture of MEC (Section II). This is done by providing an overview of relevant contributions to typical issues such as which content to cache and where, as well as which architectural aspects should be taken into consideration when looking at MEC-enabled mobile network. Section III presents details on architecture, protocol stack and algorithm logic used when implementing the testbed. This is based on the open-source Open Air Interface (OAI) [5] project. The aim of OAI is to provide a virtualized platform for research and development of 5G-related solutions. Section IV details the metrics used for performance evaluations and presents the results of the tests carried out on the testbed, in terms of reduced latency, as well as reduced load on the BHL network, on elements in the Evolved Packet Core (EPC) and beyond. Finally, Section V concludes this work and provides an outlook for the usage of its results.

II. EDGE CACHING IN THE LITERATURE

The topic of caching content at the edges of the mobile network has been investigated from multiple perspectives. These works will be briefly presented in this section to provide a more comprehensive picture for the following work. In this context, it is common to refer to the concept of Content-Centric Mobile Networks (CCMNs) as caching becomes a prominent criteria in the network design process, intertwined with more typical ones like communications-related and computing-related considerations, as pointed out by Andreev *et al.* [6].

By equipping network elements with cache capabilities, two fundamental questions arise: which content should be cached and where. The former entails estimation of the popularity of the content where several solutions have been proposed, as, for example, the use of Extreme Learning Machine (ELM) in [7].

Optimal caching content placement has been investigated in [8]—among others—with a specific focus on proactive location selection based on User Equipment (UE) mobility patterns.

More specifically, regarding the architecture of MEC systems, Tang and Quek [2] point out that the role edge caching plays in mobile networks is ultimately the evolution of a need to progressively move the content closer to its consumer. As such, the use of CDNs as Core Network (CN) caches leads to the usage of BS caches in the RAN in the context of Information-Centric Networking (ICN).

In diversified RAN architectures, such as C-RAN, it becomes also relevant to exploit a further intermediate location for storage, i.e. the BBU pool as noted in [2]. This

is considered as a synthesis of the shortcomings of the other two types of locations.

To the best of the authors' knowledge, the work presented in the following is the first one of its kind, focusing on practical implementation and quantification of user-side metrics, to evaluate possible improvements for the QoE. It is, however, possible to draw a comparison with similar works focused on simulation of a MEC deployment. Tran and Pandey [9] for example, present simulation results on the performances of a collaborative MEC algorithm in improving the content access delay, among others.

The setup described in the following has been designed to be transparent to these different approaches of designing the RAN, so that it is possible to build on top of the works presented here to give insights into the framework of tools needed to deploy MEC solutions in both scenarios.

Finally, it is relevant to highlight some practical methods used for locating where the desired content is stored. In Section III-D below, two possible solutions are presented, namely Domain Name System (DNS) and Application Layer Traffic Optimization (ALTO) [10].

III. SOLUTION OVERVIEW

The solution presented in this paper has a strong inclination toward being practically implementable with openly-available tools and components. The proposed base-station caching solution is built on top of the reference design of the LTE networks. The following sections briefly review the relevant parts of architecture and protocols of an LTE system. The section then moves on to the proposed MEC architecture, the BS protocol stack, and the logic used to locate the content.

A. LTE ARCHITECTURE SUMMARY

A conventional LTE network consists of two main sections called EPC and Evolved-Universal Terrestrial Radio Access Network (E-UTRAN), interconnected by the aforementioned network segment called BHL [11]. The main functional elements in the EPC are the Mobility Management Entity (MME), the Serving Gateway (S-GW), and the Packet Data Network Gateway (PDN-GW). While the MME handles the signalling between the UE and the CN, it is up to the two gateways to handle data in the user plane. More specifically, all Internet Protocol (IP) traffic flows through the S-GW, which is thus used—among other tasks—as a user plane data anchor while UEs switch between different Evolved Node Bs (eNodeBs). Additionally, because of such prominence, the S-GW is also used for Legal Interception (LI) of user traffic [11].

The PDN-GW—on the other hand—is responsible for assigning IP addresses to the UEs, Quality of Service (QoS) enforcement and flow-based charging, thus implementing the Policy and Charging Enforcement Function (PCEF) [11].

Additional nodes are needed besides these core ones in the EPC. In particular, the Policy and Charging Rules Function (PCRF) and the Home Subscriber Service (HSS). The former stores the aforementioned charging and shaping rules with Traffic Flow Templates (TFTs) used by the PDN-GW. The latter holds UE subscription data with possible restrictions, Access Point Name (APN) details and the UE-MME associations. Data between external networks and the UEs are encapsulated and transported by means of Evolved Packet System (EPS) bearers [11].

In the E-UTRAN instead, the only nodes present are the eNodeB and the UE. In the typical LTE user plane, the eNodeB acts as a bridge node between the E-UTRAN and the CN. A relevant portion of the LTE user plane protocol stack is shown in Fig.2.

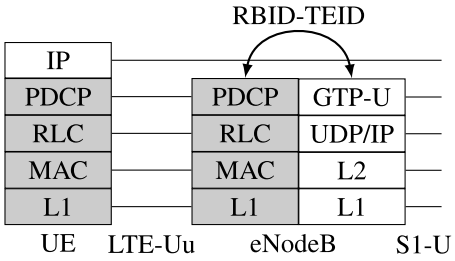

FIGURE 2. UE and eNodeB protocol stack. Shaded parts indicate protocols operating in E-UTRAN.

In conventional operations and once the bearer is set up, the eNodeB works as a gateway. In downlink, for example, the eNodeB receives IP packets on the S1 interface from the EPC marked with a GPRS Tunneling Protocol (GTP) S1-Tunnel Endpoint IDentifier (TEID). It then finds a mapping between the TEID and the Radio Bearer IDentifier (RBID) used to identify the radio bearer and finally delivers the packets toward the LTE-Uu interface.

This is then reversed in the uplink. IP packets received over the EPS Radio Access Bearer (E-RAB) from the LTE-Uu interface are encapsulated at the Packet Data Convergence Protocol (PDCP) layer. This is done using the GTP User (GTP-U) protocol and packets are assigned a TEID based on the RBID $\leftrightarrow$ TEID mapping. User data to the corresponding S-GW is then encapsulated in GTP-U packets and labeled with the TEID assigned for the bearer.

B. EDGE CACHING IMPLEMENTATION

An overview of the logical entities of the proposed edge-caching solution is displayed in Fig. 3a. In the proposed architecture, a content cache (e.g., a web-server serving web pages) is co-located with the eNodeBs. If a UE data request can be served from the cache, IP packets are routed toward the cache server instead of the S-GW. Implementation details on IP packets processing in the eNodeB are presented in Section III-C.

Positioning the cache servers in the eNodeB changes the user data path, so that IP packets routed from the eNodeB toward the cache server no longer traverse the S-GW and the PDN-GW. Due to this change, it is important to consider what impact this has on the functions performed by those elements now omitted. The only affected function in the S-GW is LI, which can, however, be performed directly in the cache

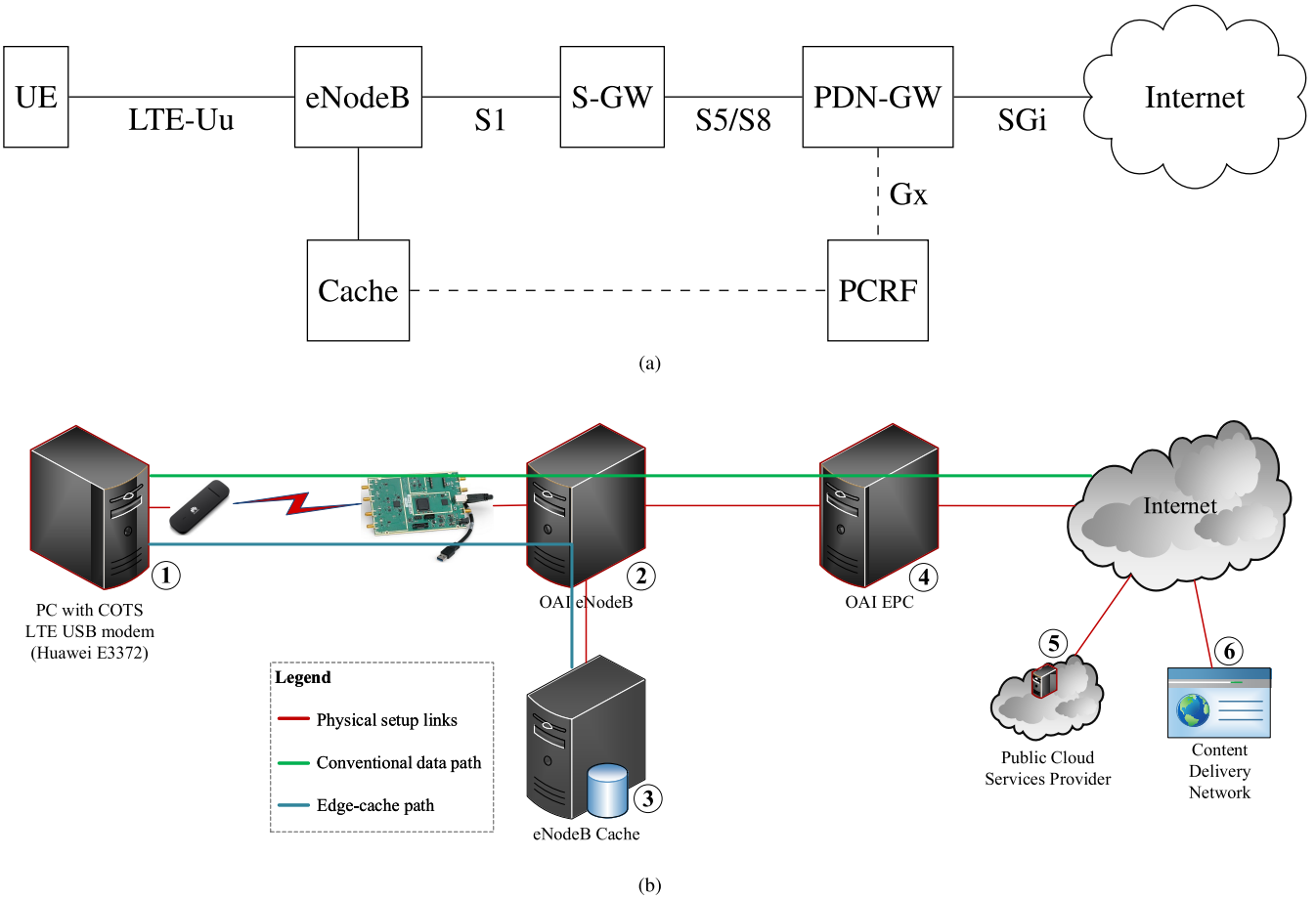

FIGURE 3. The logical architecture (3a) of the proposed edge-caching solution and the physical components (3b) used to implement the system prototype.

server. The traffic shaping performed by the PDN-GW on the EPS bearers is also affected. In order for the cache server to provide this service, communication with the PCRF is necessary via the Gx interface, as well as the implementation of the PCEF functionalities.

C. PROTOCOL STACK

Starting from the standard user plane protocol stack described previously, a number of modifications are needed when equipping the eNodeB with a local cache. In the proposed MEC solution, the eNodeB has to not only act as a gateway, but also as a router. It has to inspect the traversing IP packets and, if the conditions are met, it has to divert the packets toward the cache server instead of the S-GW.

Moreover, a conventional eNodeB is transparent to the IP packets carried. However, to enable edge caching, this needs to change, as IP destination information is needed to properly route packets either locally to the cache or toward the EPC. In the proposed solution, the flow of the packets through a PDCP/GTP-U relay includes an additional step to inspect the destination IP address.

If the address is within a predefined range assigned to the cache servers, then the packet is sent to the local cache instead of the S-GW. In the proposed solution, the interface between the eNodeB and the local cache is also GTP-U-based. This minimizes the changes needed in the eNodeB, as from its perspective the interface to the local cache is like the interface to the S-GW. On the cache-server side, the GTP-U gateway in the testbed is implemented in the Linux Operating System (OS) kernel, allowing to use cache server integrating GTP-U and web-server functions (see Fig. 4).

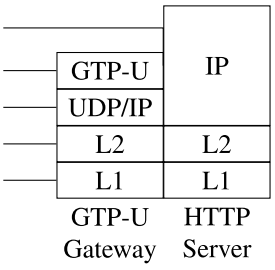

FIGURE 4. Protocol stack in the cache server with an integrated GTP-U gateway and HTTP server.

Additionally, this MEC solution is IP-version agnostic and requires only a small extra processing time per packet in the

eNodeB. To determine if a packet should be sent toward a local cache, the eNodeB performs checks on the IP version and destination address. A local cache can be made of several servers, but as long as all server addresses are within a single Classless Inter-Domain Routing (CIDR) block, it is enough to perform a single logical operation to determine where to route the packet.

D. CONTENT LOCATION AND NAME RESOLUTION

The previous section described the part of the caching solution that is responsible for data transport between the UE and the content cache. However, for it to work, UEs must know the IP address of a node that will fulfill the request. This address might be of a cache server, or of some other server if data is not available in the cache. The process of discovering the address of a network node that will serve the content request is part of the process of name resolution and routing and is described in this section.

Following ICN concepts, information that a UE tries to access, such as a web-page or a document, will be referred to as a Named Data Object (NDO) [12]. The discovery of NDO names is user-application specific and outside the scope of this work. However, once the NDO name is known, two possible methods for resolving it to an IP address are described in the following.

For NDOs identified using Uniform Resource Locators (URLs), a name resolution based on DNS can be used. In this case, a DNS server must be operated by the mobile operator. Upon reception of the client's DNS query, the server will perform two actions. First, it will query the MME to know the eNodeB that the client is attached to. Then, if the eNodeB in question contains a cache server, the cache server will be checked for the required content. If the cache server has the required content, the IP address of the cache server will be returned. Alternatively, the DNS server will perform a recursive DNS query and return the result to the client.

However, DNS-based NDO name resolution has several shortcomings. First, to optimize cache space usage, an operator might have different content available in different cache servers. This implies, that the NDO name resolution has to be performed by a DNS server having knowledge of all content available in edge-caches. This rules out cases where clients use different DNS servers than those deployed by the operator, as they do not contain such information. Second, to increase domain resolution speeds, the OS might choose to cache the DNS replies. However, the cached address is valid only as long as the UE is connected to the same eNodeB and the content is still available in the edge-cache.

As an alternative to the DNS-based content location, a NDO name can be resolved using the ALTO protocol [10]. The goal of the ALTO protocol is to provide a framework that an operator can use to expose information about its network. ALTO specifies the structure and the syntax of the messages exchanged between the clients and the ALTO server. The generic use case of ALTO to discover the closest CDN node is introduced in [13]. The rest of the section describes how ALTO can be used with the proposed MEC solution.

To leverage ALTO for NDO IP address discovery, the ALTO server assigns a unique Provider-defined Identifier (PID) to each NDO. A PID is an alphanumeric identifier created by the operator to identify a network location. In this context a network location is one or more IP addresses used by the cache servers containing the requested NDO. Upon receiving an ALTO protocol request to locate the PID, an ALTO server would perform the same two tasks as a DNS server described previously. The results are then returned to the UE in the form of an ALTO *Network Map*. The returned network map contains a list of PIDs, each with an IP address of a cache server. Using the ALTO-provided network map to discover NDO IP addresses has several advantages. First, it allows the operator to inform the UE about all cache servers that contain the required NDO in single operation. Second, the UE can use additional ALTO protocol features (such as cost-maps) to inquire which cache-server is preferred in terms of data routing costs (i.e., available bandwidth, network load and others).

IV. PERFORMANCE EVALUATION

In order to validate the proposed system, a prototype MEC implementation was evaluated for correctness and performance. The goals of the evaluation were two-fold. First, the tests were used to prove that the testbed worked as intended. This was achieved by ensuring that IP packets were routed in the eNodeB toward the cache when required. Second, a performance comparison was performed to quantify the improvements in terms of parameters impacting the user-perceived QoE. The overall network responsiveness was quantified by measuring the ping response times. On mobile devices, about 20% of all data traffic is used for web-browsing [3] delivered over the HTTP protocol. The improvement in web-browsing performance was measured by observing the HTTP response rate. Finally, about 60% of all mobile data traffic is currently used to deliver streaming audio and video [3]. The improvement in streaming multimedia performance was measured by observing the average User Datagram Protocol (UDP) packets jitter. The performance of the proposed system in ping and HTTP tests was compared to that of testing against cloud and CDN servers—two common methods of serving and also caching data in the Internet.

A. PERFORMANCE METRICS DEFINITION

The performance of the system prototype was tested using the following configuration. All tests were carried out using the reference implementation shown in 3b. The S1 interface link delay between the eNodeB and the EPC was set to 5 ms, which is considered as a realistic estimate of the delay over the S1 interface of a real network. A computer running Linux Ubuntu 16.04 with a Huawei E3372 LTE dongle was used as UE. For over-the-air testing, the LTE-Uu interface was set to a bandwidth of 10Mhz (50 Resource Blocks).

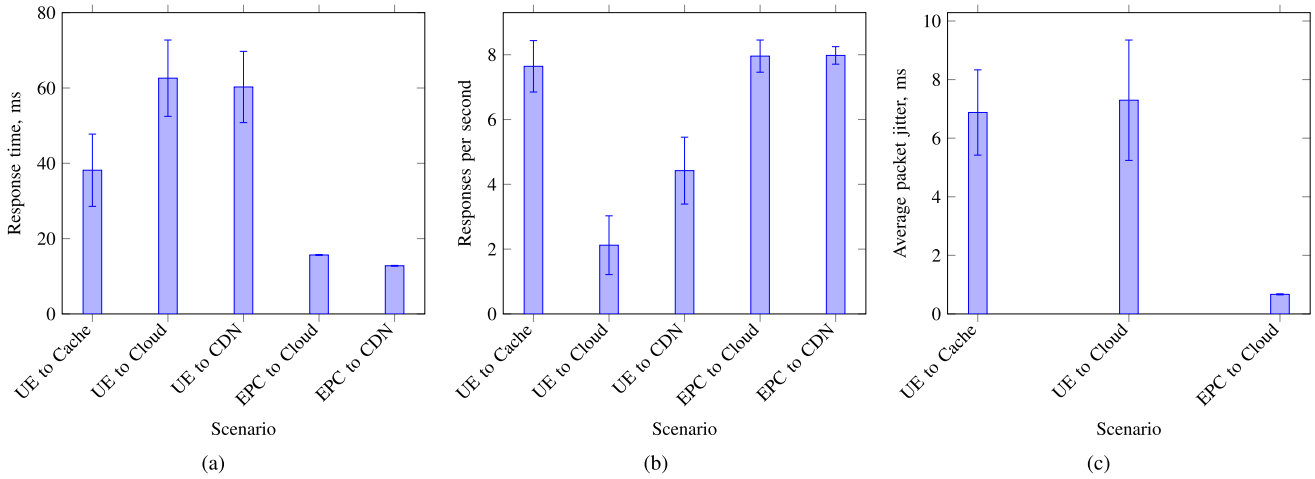

FIGURE 5. Test results of prototype edge-caching solution. UE - User equipment, EPC - Evolved Packet Core, CDN - Content Delivery Network. Error bars indicate one standard deviation. (a) Average ping response time. (b) Average number of HTTP responses. (c) Average UDP packet jitter.

Ping tests were performed by sending 100 consecutive *ping* requests between the computer with the Commercial Off-The-Shelf (COTS) LTE modem (UE, node 1 in Fig. 3b) and a cache server (node 3 in Fig. 3b) connected directly to the eNodeB (node 2 in Fig. 3b), a server in a public cloud (node 5 in Fig. 3b), and a CDN server (node 6 in Fig. 3b). The same test was also repeated from the EPC (node 4 in Fig. 3b), however without pinging the cache server co-located with the eNodeB.

The HTTP performance tests were carried out using the *httperf* software. During the tests, a web server distributing a 500 KB file was used. The size of the payload was derived by taking the average size of Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS) content as observed among the 100 most popular Internet websites [14]. Each test consisted of 1500 requests for such file, with a new Transport Control Protocol (TCP) connection created for each request. The concurrent test load in each case was discovered by gradually increasing the number of concurrent connections, until the client connections started to time-out. The discovered value was later used to repeat the whole test. The indicated test results show the average number of responses per second as observed in 5-second measurement windows. Performing 1500 requests allowed to collect enough 5-second measurement windows to derive statistically valid results.

The UDP packet jitter tests were carried out using *iPerf3* software. The offered data load was 25 Mbps in order to fully saturate the connection. The actual jitter calculation follows the description in [15]. As this test requires test software to be run on both connection ends, jitter tests were not carried out between the UE and the CDN.

The results presented here were obtained by using a single UE connected to the eNodeB without any traffic shaping in effect. The performance in the real mobile networks will depend, among other parameters, on the number of UEs connected to the same eNodeB, on the scheduling algorithm in the eNodeB, on the traffic shaping parameters and on the channel conditions in the LTE-Uu interface among other factors.

B. RESULTS AND DISCUSSION

The evaluation of the MEC solution focused on both artificial and real-world usage tests. Fig. 5a shows the average ping time from the UE to the edge-cache server, a server hosted in the public cloud and a server in the CDN. Ping measurements from the EPC to the servers in the public cloud and the CDN are included to have a reference delay once packets traverse the FHL and BHL networks. Using the prototype implementation, the ping time from the UE to the edge-cache is 39% (24.45 ms) and 37% (22.11 ms) shorter compared to the ping time to the cloud and CDN servers. These results can be adjusted to remove the influence of the BHL delay (5 ms) and the delay from the EPC to the remote servers. By taking these adjustments, the time savings from the UE to the server in the public cloud is 10% (3.78 ms) and 11% (4.32 ms) to the CDN server.

Fig. 5b shows the average number of HTTP responses received per second from the edge-cache and the servers in the cloud and in the CDN. This represents a real-life test, as an increase in the number of HTTP responses per time window leads to a shorter application response time, which ultimately improves the perceived web-browsing QoE. The results indicate that by serving content from the edge-cache, the response rate can be increased 3.6 and 1.73 times, compared to the cloud and CDN servers, respectively. The figure also shows that the response rate in the edge-cache case approaches that of accessing remote servers from the EPC. A big difference in the performance can partly be explained by the shorter response times between the UE and the edge server compared to Cloud and CDN use-cases. As TCP connection establishment, tear-down, and data transfer require multiple data round-trips, a shorter delay to the edge-cache server corresponds to a higher overall response rate.

The quality of real-time streaming content is affected by the average packet jitter [15]. Fig. 5c shows the average observed jitter from the UE to the edge-cache and the server in the cloud. As shown in Fig. 5c, the use of edge-cache provides a marginal improvement of 6% (0.42 ms). This also shows that most of the jitter is then induced on the LTE-Uu interface between the UE and the eNodeB.

The solution presented can thus be considered as a viable one for improving QoE for mobile users and has the benefit of being simple to implement as it requires minor and well scoped software changes in the eNodeB. Putting it into perspective, it constitutes a core building block for a Caching-Oriented Vertical Framework to Enable Fog Elasticity (COVFEFE) where the presented implementation can be repeated and scaled in several different network architectures.

V. CONCLUSIONS

MEC solutions propose the use of content caches located in the base stations (eNodeBs) to serve popular content to the UEs. Co-locating the cache server with the eNodeB offers several advantages. Besides improving QoE for the UEs, it also reduces the load on the operators' BHL network. A prototype of a MEC system has been presented in this paper. It focuses on minimizing the changes needed in order to implement it in a real network. In order to verify the design of the proposed edge-caching system, the prototype has been implemented and tested using a COTS UE.

The results obtained show that the data delay between the UE and the eNodeB-cache is lower compared to the delay between the UE and the remote servers. This in turn has a positive influence on HTTP performances. The test results indicate that the number of HTTP responses per second increases between 1.7 and 3.6 times when served from the edge-cache. Other communication protocols requiring multiple round-trips of data would be positively affected as well due to the reduced data delay.

The tests also show that the proposed system offers only a marginal improvement in terms of the reduced packet jitter. As the most jitter is induced in the LTE-Uu interface between the UE and the eNodeB, co-locating content in the edge-caches will not offer significant reduction of jitter. However, while the jitter is not reduced, placing jitter sensitive content in the edge-caches would still reduce the load on the operators' BHL networks.

REFERENCES

[1] J. Kakar, S. Gherekhloo, Z. H. Awan, and A. Sezgin, "Fundamental limits on latency in cloud-and cache-aided HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[2] J. Tang and T. Q. S. Quek, "The role of cloud computing in content-centric mobile networking," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 52–59, Aug. 2016.
[3] *Ericsson Mobility Report*, Ericsson AB, Stockholm, Sweden, Jun. 2017.
[4] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.
[5] *OpenAirInterface: 5G Software Alliance for Democratising Wireless Innovation*. Accessed: Jul. 11, 2017. [Online]. Available: http://www.openairinterface.org/
[6] S. Andreev *et al.*, "Exploring synergy between communications, caching, and computing in 5G-grade deployments," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 60–69, Aug. 2016.
[7] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach," *IEEE Access*, vol. 5, pp. 5870–5881, 2017.
[8] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.
[9] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. 13th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS)*, Feb. 2017, pp. 165–172.
[10] R. Alimi *et al.*, *Application-Layer Traffic Optimization (ALTO) Protocol*, document RFC 7285, RFC Editor, Fremont, CA, USA, Sep. 2014, pp. 1–91. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7285.txt
[11] S. Sesia, M. Baker, and I. Toufik, *LTE-The UMTS Long Term Evolution: From Theory to Practice*. Hoboken, NJ, USA: Wiley, 2011.
[12] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.
[13] M. Stiemerling, S. Kiesel, M. Scharf, H. Seidel, and S. Previdi, *Application-Layer Traffic Optimization (ALTO) Deployment Considerations*, document RFC 7971, RFC Editor, Fremont, CA, USA, Oct. 2016, pp. 1–77. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7971.txt
[14] *HTTP Archive—Trends*. Accessed: Jul. 11, 2017. [Online]. Available: http://httparchive.org/trends.php?s=Top100
[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, document RFC 3550, RFC Editor, Fremont, CA, USA, Jul. 2003, pp. 1–104. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3550.txt

JUSTAS PODERYS received the B.Sc. degree in telecommunications engineering from Vilnius Gediminas Technical University, Lithuania, and the M.Sc. degree in telecommunications engineering from the Technical University of Denmark in 2014, where he is currently pursuing the Ph.D. degree. His areas of research are multimedia streaming over heterogeneous networks, peer-to-peer communication, and data routing.

MATTEO ARTUSO received the B.Sc. degree in information engineering from the University of Padua, Italy, in 2012, and the M.Sc. degree in telecommunication engineering from the Technical University of Denmark in 2014, where he is currently pursuing the Ph.D. degree. He is currently the Chief Technology Officer with Unumed. He is also the Chairman of the local IEEE Student Branch. His main research areas are architectures and protocols for next-generation mobile networks, with a special focus on cloud networks, virtualisation, and self-organisation.

CLAUS MICHAEL OEST LENSBØL received the B.Sc. degree in network technology and IT from the Technical University of Denmark in 2016, where he is currently pursuing the M.Sc.Eng. degree in telecommunication. He is the Chief Technology Officer with Sowa, the wireless speaker startup. His research interests include synchronization in mobile networks, protocol design, and telecommunication software, with a growing interest of virtualization of networks and services in networks.

HENRIK LEHRMANN CHRISTIANSEN received the M.Sc.E.E. and Ph.D. degrees in telecommunications from the Technical University of Denmark. He is currently an Associate Professor in mobile communication with the Technical University of Denmark. He also has several years of experience with the telecom industry. His main areas of research are mobile network architectures and planning, especially fronthaul networks for cloud-radio access networks and evolution towards 5G mobile.

JOSÉ SOLER (SM'17) received the M.Sc. degree in telecommunication engineering from Zaragoza University, Spain, in 1999, the M.B.A. degree from UNED in 2016, the Ph.D. degree in electrical engineering from DTU, Denmark, in 2005, and the Diploma degree in management from Erhversakademiet Copenhagen Business in 2010. He was with ITA, Spain; with ETRI, South Korea; with COM DTU, Denmark; and also with GoIP International, Denmark. He is currently an Associate Professor with the Networks Technology and Service Platforms Group, DTU Fotonik, Technical University of Denmark. His research interests include integration of heterogeneous telecommunication networks and telecommunication software and services. He has been participating in applied-research projects since 1999. He serves as a TPC Member in an extended number of conferences and journal review panels.